Problem 1.

Given a strongly-connected directed graph $G = (V, E)$, define the shortest path distances from $u$ to $v$ as $d(u, v)$ for all $u, v \in V$. Then, we can formulate the given problem as a linear program as follows.

$$
\begin{align*}
\text{maximize} & \quad \sum_{u, v \in V} d(u, v) \\
\text{subject to} & \quad d(u, u) = 0, \forall u \in V \\
& \quad d(z, y) \leq d(z, x) + w(x, y), \forall (x, y) \in E, \forall z \in V
\end{align*}
$$

We need to maximize each $d(u, v)$ in the objective function since the shortest path distance from $u$ to $v$ is the maximum value satisfying the triangle inequalities, as described in the textbook. Having the maximum values of each $d(u, v)$ is equivalent to having the maximum value of the sum of all of those values, so we sum up all of $d(u, v)$ in the objective function.

Problem 2.

Given an undirected flow network $G = (V, E)$, we construct the directed flow network $G' = (V, E')$ by having two antiparallel directed edges $(u, v)$ and $(v, u) \in E'$ for each edge $(u, v) \in E$. The capacities of these new directed edges are $c(u, v)$, the capacity of $(u, v) \in E$.

Two flow networks $G$ and $G'$ produce the same flow. If the undirected edge $(u, v) \in E$ has a flow, $f(u, v)$, from $u$ to $v$, it is equivalent to the flow from $u$ to $v$ in $G'$ with a value of flow $|f(u, v)|$. Conversely, if there are $f(u, v)$ and $f(v, u)$ in $G'$, then it is same as the flow on $(u, v) \in E$ with a value of flow $|f(u, v) - f(v, u)|$.

1) Construct the directed flow network $G'$ as described above, and run Edmonds-Karp algorithm to find a maximum flow from $s$ to $t$. This will give us the same maximum flow in $G$. The running time of this algorithm is $O(VE^2)$.

2) Since $c(u, v) = 1$ for all $(u, v) \in E$, the directed flow network $G'$ constructed will produce a maximum flow $f_{\text{max}}$, such that $|f_{\text{max}}|$ is in fact the number of edges crossing a minimum cut by the Max-Flow Min-Cut theorem. In order to reduce a value of the maximum flow as much as possible by deleting $k$ edges, we
just choose $k$ edges among all edges crossing a minimum cut. Since a minimum cut $(S,T)$ is a cut such that $S = \{ v \in V | v$ is reachable from a source $s \in G_f' \}$, where $G_f'$ is the final residual network, and $T = V - S$, we run BFS on $G_f'$ to find a minimum cut. Now, we know which edges crossing a minimum cut, so just choose $k$ edges among them to delete. Its running time is $O(VE)$ since $|f_{\text{max}}| = O(V)$.

**Problem 3.**

Given an undirected graph $G = (V,E)$, construct the directed graph as described in Problem 2. Select any vertex $s \in V$ as a source vertex, and find the maximum flows for $|V| - 1$ sinks $t \in V - \{s\}$. We take a maximum flow with the minimum value among $|V| - 1$ maximum flows, and find the corresponding minimum cut, as explained in Problem 2.

$|V| - 1$ pairs of $(s,t)$ are enough to consider since $s$ must be placed in one of partitions of a minimum cut, and the flow network for each $t$ will examine the case when $t$ is in the opposite partition.

The running time is $O(V^2E)$ since each maximum flow algorithm will run in $O(VE)$, as described in Problem 2, and there are $O(V)$ flow networks.